



Universidade de Brasília

Modelo de Gerenciamento de Produtos Digitais para Startups

João Moraes Simões

Brasília, 2019



Universidade de Brasília

Modelo de Gerenciamento de Produtos Digitais para Startups

João Moraes Simões

Orientação

Symone Jardim

Brasília, 2019

AGRADECIMENTOS

Em primeiro lugar gostaria de agradecer à toda a equipe do Ribon por trabalhar de forma dedicada e apaixonada, e pela abertura que proporcionaram para realização deste projeto. Por ser uma equipe incrível e muito particular, com quem tenho o prazer de trabalhar todos os dias.

Agradeço à minha família pelo apoio e compreensão durante toda a realização do projeto, e aos meus amigos pelos momentos de descontração e ajuda.

Agradeço à todos os profissionais e empresas que contribuíram para o projeto me ajudando com toda a experiência e momentos de mentoria. Obrigado pela abertura em compartilhar suas histórias.

Finalmente, agradeço à professora Symone pela energia inspiradora, paciência com um startupeiro distraído, e por ser a professora Symone. Quem sabe, sabe.

RESUMO

O objetivo deste projeto é desenvolver um modelo de gerenciamento de produtos digitais para startups. O projeto foi desenvolvido utilizando a startup Ribon como estudo de caso, e o resultado final foi aplicado à startup, sendo possível compreender como as soluções propostas impactam na gestão de uma empresa real. O projeto foi desenvolvido por meio de extensa pesquisa, contato com profissionais da área e com a própria equipe do Ribon.

Palavras-chave:

Gerenciamento de produto, startup, produto digital, pesquisa.

LISTA DE ILUSTRAÇÕES

Figura 1 - Tela de "histórias", a tela inicial do app onde usuários recebem notícias e histórias positivas todos os dias.....	09
Figura 2 - Visualização de um anúncio na tela de histórias.....	09
Figura 3 - Tela de doação, onde é possível escolher entre 4 causas sociais disponíveis para receber doações.....	10
Figura 4 - Tela de Perfil, onde é possível acompanhar o histórico de impacto das doações.....	10
Figura 5 - Double Diamond.....	14
Figura 6 - Percepções da equipe de desenvolvimento registradas através de um mapa de empatia.....	25
Figura 7 - Percepções da equipe de <i>growth</i> registradas através de um mapa de empatia.....	25
Figura 8 - Agrupamento das idéias e <i>insights</i> obtidos por meio dos mapas de empatia.....	28
Figura 9 - Etapa de Pré-Projeto.....	31
Figura 10 - Formulário de inicial para cadastro de uma funcionalidade nova no Pipefy.....	32
Figura 11 - Etapa de Design.....	33
Figura 12 - O projeto de novas funcionalidades normalmente começa com desenhos de baixa fidelidade feitos em papel.....	34
Figura 13 - Figma, software utilizado para confecção dos leiautes.....	35
Figura 14 - Etapa de Desenvolvimento.....	36
Figura 15 - Etapa de Avaliação.....	37
Figura 16 - Primeira versão do <i>Roadmap</i>	38
Figura 17 - Segunda versão do <i>Roadmap</i>	40
Figura 18 - Antes do projeto o <i>Handoff</i> era feito de forma desestruturada, onde as telas não eram organizadas formalmente, e não havia nenhuma descrição contextual.....	38

Figura 19 -Mapeamento de fluxo de usabilidade em um documento de <i>hand-off</i>	42
Figura 20 - Job Stories em um documento de <i>Hand-off</i>	43
Figura 21 - O documento de <i>handoff</i> completo é composto por uma parte inicial com Job Stories que pode informar um ou mais possíveis fluxos de usabilidade, que são mapeados no documento.....	44
Figura 22 - <i>Kanban</i> de acompanhamento de funcionalidades no Pipefy.....	45
Figura 23 - Páginas do Documento Central de Produto.....	47

SUMÁRIO

1. INTRODUÇÃO.....	07
1.1. Justificativa.....	07
1.2 Ribon.....	08
1.3 Objetivos.....	11
1.4 Processo de design.....	12
2. A PRÁTICA PROJETUAL EM STARTUPS.....	13
2.1 Double Diamond.....	14
2.2 Entrega de Valor em Produtos Digitais.....	15
2.3 Composição de Equipes de Produto.....	17
2.4 Governança Horizontal e Autogestão.....	18
3. PESQUISA QUALITATIVA.....	20
3.1 Entrevistas com Especialistas.....	20
3.2 Imersão.....	22
3.3 Entrevistas com a Equipe do Ribon.....	24
4. PROPOSTA DE SOLUÇÃO.....	30
4.1 Panorama Geral do Modelo de Gerenciamento de Produto do Ribon	30
4.2. Detalhamento de Artefatos no Modelo de Gerenciamento de Produto	38
5. CONSIDERAÇÕES FINAIS.....	48
REFERÊNCIAS BIBLIOGRÁFICAS.....	50
ANEXOS.....	51

1. INTRODUÇÃO

Uma startup é uma instituição projetada para entregar um novo produto ou serviço sob condições de extrema incerteza (RIES, 2011). É difícil dizer com exatidão o momento em que empresas desse formato surgiram: diferentes autores têm definições variadas para o que seria uma startup, e portanto diferentes pontos de origem para esse tipo de empresa, que chega a datar da grande depressão de 1929 ou antes.

O fato é que o surgimento das startups como conhecemos hoje está ligado principalmente ao advento da globalização, à indústria de tecnologia da informação e consequentemente ao vale do silício na Califórnia. Foi nesse contexto, principalmente à partir da década de 1970, que surgiram empresas como Apple, Microsoft e Google, assim como novas práticas e modelos de gestão mais focados em inovação e agilidade sempre de forma intimamente ligada aos avanços tecnológicos.

As práticas e a mentalidade que esses pioneiros estabeleceram ao longo das últimas décadas do século XX continuaram se desenvolvendo por meio de outras empresas que surgiram em seguida, até se tornarem os modelos de gestão e projeto de produto vistos em startups atuais.

Um dos eventos importantes nessa caminhada foi a publicação do Manifesto Ágil, em 2001, que estabelecia boas práticas para o desenvolvimento de produtos no contexto de tecnologia e inovação, e foi escrito de forma independente por profissionais proeminentes da área de tecnologia na época. O Manifesto Ágil se mantém, com algumas ressalvas, extremamente atual. Alguns dos princípios apresentados oferecem *insights* importantes que ainda são representativos da visão de startups até hoje.

Esses e outros princípios auxiliam no entendimento da mentalidade por trás de startups bem sucedidas, e orientam equipes de produto de startups a serem extremamente responsivas a mudanças de diversas naturezas, aprenderem rapidamente por meio do contato com usuários, e consigam manter uma visão estratégica sobre o que estão projetando.

1.1 Justificativa

Uma das novas competências que surgiu com o ecossistema de startups é o Gerenciamento de Produto, tipicamente liderado por um Gerente

de Produto (GP). Segundo Ben Horowitz (1997) o Gerente de Produto é o responsável pela estratégia, evolução e gestão do produto de uma startup. Normalmente o GP trabalha gerenciando uma equipe principalmente formada por designers e desenvolvedores, e alinhado com necessidades dos usuários busca desenhar a melhor estratégia para o desenvolvimento do produto, sempre de maneira iterativa. De forma geral, o gerenciamento de produto representa a prática projetual dentro de uma startup, com um foco no âmbito estratégico.

Existem diversas abordagens para se fazer gestão de produto, cada uma com ênfases diferentes. Por isso, cada startup trata gestão de produto segundo seu próprio modelo, dependendo das necessidades e momento da empresa, composição da equipe de produto, etc. Esse projeto aborda esse desafio comum de startups, o gerenciamento de produto, utilizando como estudo de caso o Ribon, uma startup da qual sou fundador e onde trabalho até hoje.

1.2 Ribon

O Ribon (Disponível em: ribon.io) é uma startup de impacto social fundada em 2016 por mim em conjunto com dois outros sócios, e seu objetivo é construir uma plataforma digital de experiência de doação. Em sua forma atual, o Ribon é um app para aparelhos Android ou iOS que permite que usuários façam doações diárias de forma descomplicada e digital para comunidades em situação de extrema pobreza.

O app funciona dessa forma: usuários realizam o *download* do app e tem acesso à uma seleção de boas notícias diariamente (figura1).



Figura 1 Tela de "histórias", a tela inicial do app onde usuários recebem notícias e histórias positivas todos os dias.

Ao ler essas notícias usuários visualizam anúncios de empresas patrocinadoras que pagam para ter sua marca visível no app (figura 2).

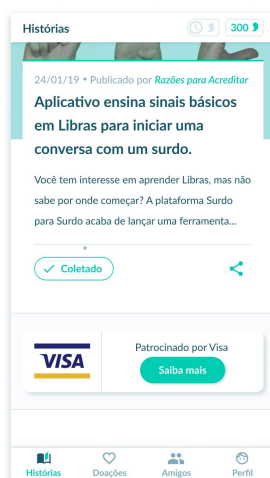


Figura 2 - Visualização de um anúncio na tela de histórias.

O dinheiro proveniente desses anúncios é dividido entre os usuários na forma de "ribons", pontos ou moedas virtuais que podem ser acumuladas e doadas para qualquer uma das 4 instituições de impacto social parceiras da startup (figura 3).



Figura 3 -Tela de doação, onde é possível escolher entre 4 causas sociais disponíveis para receber doações.

Usuários podem acompanhar o impacto de suas doações por meio de um histórico de impacto no seu perfil (figura - 4), além de ter acesso à um relatório bimestral que presta conta sobre as doações realizadas por todos na plataforma.



Figura 4 Tela de Perfil, onde é possível acompanhar o histórico de impacto das doações.

Dessa forma usuários do Ribon conseguem apoiar ONGs internacionais altamente efetivas com doações frequentes sem ter que doar o seu próprio dinheiro, mas sim o dinheiro proveniente de anunciantes dentro da plataforma.

As instituições apoiadas pelo projeto são: Project Healthy Children, trabalhando com fortificação alimentar e nutrição básica com foco em crianças; Schistosomiasis Control Initiative, trabalhando com combate à Doenças Tropicais Negligenciadas; Evidence Action, trabalhando com tratamento de água para consumo e sanitização; e Living Goods, trabalhando com serviços e produtos de saúde básica. Todas as instituições atuam atendendo populações em situação de extrema pobreza na África e Ásia.

Durante o período em que o projeto descrito neste relatório foi desenvolvido, o Ribon saltou de menos de 10 mil para mais de 60 mil usuários mensais, que juntos fizeram mais de um milhão de doações. Esse crescimento na adoção do produto da startup aconteceu em conjunto com um crescimento da startup em si. A equipe de produto que originalmente era formada apenas por um designer (eu) e um desenvolvedor, agora é uma equipe de 13 pessoas: 6 desenvolvedores, 4 designers, uma especialista em growth, um especialista em análise de dados e um redator. Essa equipe é responsável pelo planejamento e implementação de novas funcionalidades no aplicativo, e com o crescimento recente do público, essa tarefa vem se tornando cada vez mais desafiadora e exigindo uma maior maturidade da equipe, tanto no âmbito técnico quanto no estratégico. Manter essa equipe interdisciplinar trabalhando em harmonia, com boa produtividade, e com uma boa vazão de valor para os usuários do serviço é um grande desafio.

É nesse contexto que a implementação de um modelo de gerenciamento de produto é tão importante. A empresa já tinha alguns mecanismos de gestão implementados, mas ainda precisava de um processo mais maduro e formalizado que seja de fato um processo operacional regular.

1.3 Objetivos

Por isso, o objetivo desse projeto **é implementar um modelo de gerenciamento de produto na startup**. Os objetivos específicos do projeto são:

1. Fazer um levantamento de práticas da área de gerenciamento de produto, como elas acontecem no mercado, e como podem ser implementadas no Ribon.

2. Realizar pesquisas com a equipe do Ribon para compreender os desafios e oportunidades na estruturação da área de gerenciamento do produto no contexto específico dessa startup.
3. Implementar práticas e processos de gerenciamento de produto no Ribon de acordo com os aprendizados da pesquisa e acompanhar seus resultados.

A expectativa é que a implementação de um modelo de gerenciamento de produto que atenda à essas condições aumente a produtividade da equipe e qualidade das suas entregas, e possa entregar mais valor para usuários do serviço.

1.4 Processo de design

Para atingir os objetivos, foram realizadas leituras de bibliografias de gestão em startups e gerenciamento de produto, além de pesquisas qualitativas que incluíram entrevistas com especialistas e uma imersão em uma empresa mais consolidada no mercado.

Também foi necessário entender melhor a realidade do Ribon e entender a perspectiva dos membros da equipe à respeito do gerenciamento de produto. Para isso, foram realizadas sondagens com a equipe que foram consolidadas em mapas de empatia.

O modelo proposto passou por versões diferentes ao longo da realização do projeto, e eventualmente se consolidou no modelo apresentado neste relatório.

2. A PRÁTICA PROJETUAL EM STARTUPS

Um dos desafios de qualquer startup é alcançar a chamada Validação Problema-Solução. Isso significa ter um entendimento abrangente e claro do problema que a startup se propõe a resolver, e uma proposta de uma forma excelente para solucioná-lo. Nesse sentido, a Validação Problema-Solução representa um encaixe coerente entre o entendimento do problema e uma solução inovadora que de fato atenda às necessidades de pessoas com essa dificuldade. Normalmente se considera que essa validação foi alcançada quando uma a solução proposta de fato começa a ser utilizada de forma massificada como meio para sanar o problema identificado (The Lean Startup, 2011).

Para alcançar a validação problema-solução, é necessário ter um entendimento claro do conceito de Proposta de Valor, já que ela representa a forma com que o problema será solucionado. A proposta de valor é baseada em um conjunto de produtos e serviços que entregam valor para um segmento de usuários (Value Proposition Design, 2014). Nesse sentido, a proposta de valor representa o entendimento de um problema específico, e uma proposta de solução que atenda as demandas de usuários. Por isso, ela frequentemente é descrita sob o ponto de vista dos usuários.

Esse conceito é importante para startups porque esse tipo de empresa sempre busca atender à uma necessidade de um segmento específico de mercado fornecendo uma solução inovadora.

A Proposta de Valor de qualquer startup está sempre em constante mudança à medida em que ela identifica novas oportunidades de atuação, e por isso equipes de produto estão sempre pesquisando e testando novas soluções. Isso define o caráter iterativo da prática projetual em startups: em um contexto em que um produto nunca está 100% completo, mas sim em constante evolução, testes e iterações frequentes são um foco da atuação da equipe de produto.

Trabalhando com essa mentalidade, fica claro que muitas vezes a forma mais eficiente de projetar soluções com um foco em resultados é propor uma solução que não necessariamente está perfeita, mas que já pode ser utilizada pelo seu público-alvo. Dessa forma é possível aprender com o modo com que pessoas estão utilizando a solução, e propor uma nova versão dessa solução. Esse processo acontece várias vezes até se chegar em uma solução realmente adequada. Por isso startups têm um foco tão grande na realização de testes para fazer validações, e consequentemente o processo de projeto é iterativo.

"Se você não tem vergonha da primeira versão do seu produto, você demorou demais para lançá-lo."

Reid Hoffman

2.1. Double Diamond

A forma com que esse processo de projeto iterativo se dá dentro de uma startup normalmente segue a lógica do Double Diamond (Design Council, 2006).

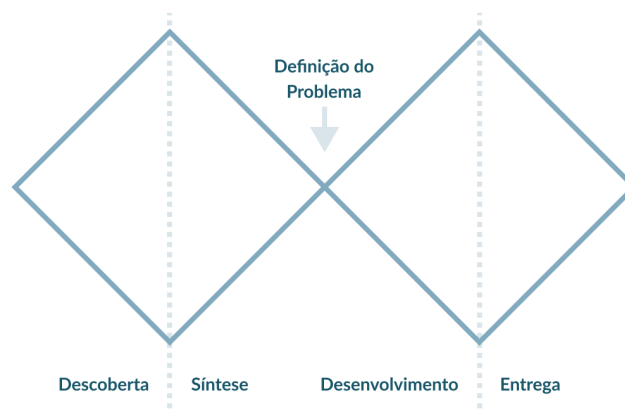


Figura 5 - Double Diamond

O Double Diamond (figura- 5) representa um processo de design não linear que tem quatro grandes fases: Descoberta, Síntese, Desenvolvimento e Entrega.

A etapa de **Descoberta** se caracteriza pela pesquisa e entendimento abrangente de um problema. Nesse momento startups normalmente realizam pesquisas tanto objetivas quanto qualitativas e acumulam informações sobre o contexto do problema que estão se propondo a resolver.

Na etapa de **Síntese**, busca-se organizar e priorizar as informações coletadas durante a etapa de descoberta. O objetivo é selecionar os *insights*

mais importantes da pesquisa de forma que eles direcionem a atuação da equipe para os aspectos mais "latentes" do problema. Essa etapa também ajuda a equipe a sintetizar a pesquisa de uma forma acessível para todos.

Durante a etapa de **Desenvolvimento** os insights identificados na etapa de Síntese vão ser abordados e soluções vão ser propostas. Nesse momento a equipe realiza processos de ideação a fim de propor soluções para o problema, sendo que essas soluções devem ser informadas pelos aprendizados adquiridos por meio das pesquisas realizadas.

A etapa de **Entrega** consiste em priorizar e testar as soluções propostas na etapa anterior. Aqui o foco é identificar a melhor forma de solucionar o problema através da prototipação, onde possíveis soluções vão ser colocadas à prova por meio de testes com usuários reais. O resultado desses testes pode servir tanto para priorizar determinadas soluções em detrimento de outras quanto para melhorar uma determinada solução. Em todas as etapas o contato com usuários reais é essencial.

Esse processo normalmente não ocorre de modo rígido dentro das equipes, mas sim orgânico. Isso significa que o processo é seguido de maneira quase instintiva, por estar muito alinhado com o que é disseminado como "boas práticas" para startups. Geralmente o papel do designer dentro da startup é estar mais consciente desse processo e ajudar a equipe a realizá-lo da melhor maneira possível. O processo também não é linear, e portanto as suas etapas normalmente não acontecem de modo perfeitamente sequencial.

Por fim, não é um processo que se segue uma vez para se chegar a uma solução perfeita. De modo coerente com o aspecto iterativo da atuação de startups, o processo do Double Diamond acontece várias vezes seguidas em diversos processos diferentes dentro de uma mesma empresa.

2.2. Entrega de Valor em Produtos Digitais

No contexto de startups que tem como principal entrega um produto digital, esse processo de projeto tem algumas particularidades. Por ser um produto digital, como um sistema *web* ou aplicativo, é extremamente fácil entregar novas versões do produto para a base de usuários. Isso significa que não é necessário ter uma grande estrutura de canais de distribuição para entregar atualizações do produto, basta produzir uma nova versão com as melhorias propostas e atualizar o produto digitalmente. Esse processo de se

liberar uma nova atualização é chamado de *release*, e é uma parte integral do processo de projeto de produtos digitais.

De maneira geral, independentemente das melhorias propostas a cada *release*, sempre é positivo manter um ritmo de atualizações constante e frequente. Isso significa que muitas vezes vale mais a pena liberar uma atualização que possui uma melhoria incompleta mais cedo, do que atrasar uma *release* até que a melhoria proposta esteja 100% pronta. Desse modo é possível receber feedbacks de usuários e entregar a melhoria completa com ainda mais qualidade na próxima *release*. Isso é coerente com a noção de que não se deve ser preciosista com as soluções propostas, mas sim colocá-las à prova o mais cedo possível.

O ritmo de *releases* que normalmente é pontuado como ideal é semanal, mesmo que cada *release* tenha apenas pequenos ajustes incrementais com relação à anterior. Manter esse ritmo semanal é extremamente desafiador. O Ribon, por exemplo, não possuía um ritmo regular de *releases*, e por isso começou a trabalhar a regularidade com uma base de duas em duas semanas. Não é o ritmo ideal, mas é melhor do que não haver regularidade.

A regularidade é importante porque mantém usuários interessados no serviço. Eles conseguem ver resultados nos seus feedbacks e entendem que o produto está em constante evolução. Isso é essencial para manter saudável uma das métricas mais importantes para qualquer produto digital: a retenção.

A retenção é uma métrica que define a taxa de retenção de usuários por um determinado espaço de tempo. Uma retenção baixa indica que por mais que usuários adquiram o produto, uma grande parte deles não se mantém interessada, e consequentemente deixa de utilizá-lo. Uma retenção alta indica que usuários que adquirem o produto continuam a utilizá-lo por mais tempo, ou seja, permanecem retidos pelo serviço por mais tempo.

É importante manter a retenção alta porque não importa o quão grande for o esforço para aquisição de novos usuários, se eles não continuarem utilizando o serviço, a base de usuários vai estar sempre diminuindo.

No fundo, o ritmo de *releases* auxilia na manutenção da retenção porque garante que o serviço sempre vai entregar valor para os usuários de forma regular e frequente, mantendo-os satisfeitos.

Esse ritmo de *releases* é um desafio que o gerenciamento de produto se propõe a manter, e também tem sua importância ressaltada quando compreende-se a natureza de incerteza e risco com a qual startups

trabalham. Em um contexto em que a estratégia de um projeto está sujeita à mudanças súbitas de acordo com novas oportunidades ou desafios, o ritmo de *releases* ajuda equipe a manter uma previsibilidade com relação a sua capacidade de entregar soluções. Uma boa equipe de produto consegue prever de forma realista quanto tempo vai demorar para realizar cada entrega, e por isso consegue priorizar seu trabalho com mais clareza.

2.3. Composição de Equipes de Produto

As equipes de produto que são responsáveis por esse tipo de projeto normalmente são interdisciplinares. Neste relatório, por motivos de brevidade, as disciplinas de uma equipe de produto de startups estão resumidas em três: Design, Desenvolvimento e *Growth*.

O principal papel dos designers dentro da equipe de produto é projetar novas funcionalidades para o produto ou revisar funcionalidades já implementadas, a fim de entregar mais valor para os usuários. Normalmente esse processo segue a lógica do Double Diamond, o que significa que os designers devem ter um contato intenso com usuários durante o seu trabalho. Isso acontece de diversas formas, mas principalmente por meio de pesquisa com usuários por meio de métodos variados e de testes de usabilidade. Testes de usabilidade são essenciais para que uma nova ideia de solução seja colocada à prova mesmo antes de estar implementada, e consiste em produzir protótipos de níveis de fidelidade diversos e testá-los com usuários reais. Tipicamente uma funcionalidade passa por várias rodadas de teste de usabilidade até que a equipe se sinta confortável em se comprometer com essa solução e implementá-la de fato. Quando essa decisão é tomada, também é papel do designer produzir os leiautes da solução de forma técnica, que possa ser facilmente interpretada pelos desenvolvedores, além de instruí-los sobre como o leiaute deve ser implementado e alinhar qualquer erro técnico que possa ter ocorrido durante o processo. Esse processo de entrega de leiaute dos designers para os desenvolvedores é chamado de **Hand-off**.

Cabe então aos desenvolvedores realizarem a visão da funcionalidade proposta implementando os leiaute por meio de uma linguagem de programação. Tipicamente isso se dá por meio de desenvolvimento do *back-end*, em que a estrutura de uma funcionalidade vai ser adicionada ao sistema geral do produto, e por meio do *front-end*, onde os desenvolvedores irão implementar o leiaute projetado pelos designers, ou seja, a parte de

interface gráfica da funcionalidade. Também cabe aos desenvolvedores registrar os eventos que irão ser acompanhados em cada funcionalidade. Os eventos são por exemplo: toques em botões ou em partes específicas da interface, visualização de certos elementos, tempo gasto em uma tela específica, etc. Isto serve para que o produto colete dados sobre sua utilização pela base de usuários, e é importante para entender como a funcionalidade está sendo utilizada. Geralmente os designers definem esses eventos com a ajuda de pessoas de *growth* antes da funcionalidade começar a ser implementada.

Por fim, os profissionais de *growth* têm o papel de acompanhar e analisar esses dados com o objetivo de identificar possíveis pontos de melhoria, principalmente no que tange à aquisição de novos usuários ou retenção de usuários que já fazem parte da base. Por meio da análise desses dados, eles também podem propor uma hipótese de uma nova funcionalidade. Em geral a atuação dessa área é baseada em dados, e por isso os designers devem sempre trabalhar próximos de pessoas de *growth* desde o início do seu processo para garantir que eles abordem um projeto com um embasamento robusto de dados sobre a utilização do produto.

Growth, Design e Desenvolvimento são 3 áreas distintas que têm suas próprias linguagens e visões a respeito de desenvolvimento de produtos, e por isso pode ser desafiador mantê-las alinhadas durante todo o processo. Um dos desafios do gerenciamento de produto é manter esse alinhamento, unificando essas 3 áreas sob um único processo de projeto.

2.4. Governança Horizontal e Autogestão

Embora a estrutura organizacional horizontal não seja uma característica de todas as startups, cada vez mais é adotada por empresas desse tipo. Startups normalmente irão buscar maneiras mais atuais de trabalhar e desenvolver projetos, e isso também se aplica à gestão, em que experimentos desse tipo são frequentes mesmo em startups maiores e mais consolidadas. Mais do que isso, há um argumento para ser feito no sentido de que equipes com estruturas horizontais produzem produtos melhores porque as tomadas de decisão não estão centralizadas em uma figura de liderança como um "chefe" ou um gerente. Em uma estrutura horizontal membros da equipe se sentem mais confortáveis para questionar processos, e a ausência de uma liderança clara ajuda a evitar tomadas de decisão enviesadas.

Se a tomada de decisão não está centralizada em um gerente, cabe ao membro da equipe se autogerir. A autogestão pressupõe que é uma responsabilidade de cada membro da equipe priorizar suas atividades com base na estratégia da startup, além de tomar decisões sem necessariamente a aprovação de um superior hierárquico, mas sim por meio do aconselhamento com outros membros da equipe que estão relacionados à decisão em questão. De acordo com o Manifesto Ágil (2001):

"As melhores arquiteturas, requisitos e designs emergem de equipes auto-organizadas."

Manifésto Ágil (2001)

O gerenciamento de produto geralmente é centralizado na figura do Gerente de Produto, que atua como um líder para a equipe de produto. No caso do Ribon, startup que serviu como laboratório para o desenvolvimento desse projeto, essa estrutura não faria sentido, já que a empresa se propõe a adotar uma hierarquia horizontal. Por isso um desafio a mais do projeto é tratar o gerenciamento de produto como um modelo ou processo que é uma responsabilidade de toda a equipe e não de uma liderança central. Para que isso seja possível é necessário que esse processo esteja claro e acessível para todos os membros da equipe. Além disso, as informações sobre o produto como dados, hipóteses, leiautes, entre outras, devem estar ao alcance de todos da equipe, independentemente da área. É essa noção que informa a necessidade desse projeto de desenvolver documentação abrangente e robusta.

3. PESQUISA QUALITATIVA

Para a realização do projeto foi importante conhecer outras realidades de projeto de produtos digitais, assim como entender internamente na própria equipe do Ribon quais eram as percepções dos membros das áreas sobre possíveis pontos de melhoria.

Para conhecer outras realidades de projetos digitais, foram realizadas duas entrevistas com especialistas da área, e uma imersão em uma startup (eduK) em um estágio mais avançado, a fim de entender na prática algumas características da área.

Para entender de modo mais claro as percepções da equipe do Ribon sobre gerenciamento de produto, foram realizadas entrevistas semi-estruturadas com os membros da equipe, e as percepções dessas entrevistas foram sintetizadas por meio de mapas de empatia.

3.1. Entrevistas com especialistas

Valdir Canado

Valdir Canado é um designer de produtos digitais e experiência de usuário. Ele é fundador da Estilingue Company, empresa que trabalha com design estratégico e desenvolvimento de software para empresas. Meu contato com o Valdir foi extremamente importante por conta de sua experiência coordenando equipes de produto grandes, trabalhando em mais de um projeto por vez.

O principal aprendizado com esse contato foi o entendimento do gerenciamento de produto como um processo: **o projeto tem etapas bem definidas em formato de "esteira" como um *kanban* por exemplo, em que a atuação de cada membro em cada uma dessas etapas é muito clara.** O processo como foi apresentado pelo Valdir é interessante porque é possível perceber como cada *insight* da fase de pesquisa caminha nessa esteira, eventualmente se tornando um avatar do serviço oferecido. Esse senso de continuidade e causalidade é muito importante no contexto de startups porque esse tipo de empresa tem um grande foco em validações. Nesse caso, se observando uma funcionalidade entregue pelo time da Estilingue, é muito claro como cada aprendizado da etapa de pesquisas se tornou uma característica do produto. É um projeto com poucas perdas de informação

durante todo o processo, e com uma documentação breve, mas objetiva no sentido de direcionar as soluções.

A equipe trabalha de forma 100% remota e por isso cada etapa da esteira descrita tem uma entrega formal muito clara. Isso acontece porque quando o designer entrega o leiaute para os desenvolvedores, por exemplo, ele não necessariamente vai ter muito tempo de discussão para alinhar detalhes pequenos. Para que isso não seja um problema a entrega acontece de uma forma padronizada por meio de todos os processos de projeto de funcionalidades, permitindo que cada membro da equipe alcance um nível de independência maior e necessite de menos momentos alinhamentos com o restante da equipe.

Toda a comunicação e acompanhamento do progresso do projeto acontece por meio do Trello, mantendo os pontos de contato entre membros da equipe objetivos. Centralizar informações dessa forma é positivo para o andamento do projeto.

Raphael Farinazzo

Raphael Farinazzo é Gerente de Produto na Involves, uma empresa florianopolitana de desenvolvimento de software com foco em SaaS (Software as a Service). A conversa com o Raphael foi importante para o entendimento de alguns conceitos fundamentais do gerenciamento de produto, já que ele possui experiência extensa na área. Como ele atua como Gerente de Produto em projetos diferentes dentro da mesma empresa, tem uma abordagem mais estratégica e menos operacional no seu trabalho, e isso se refletiu nos principais aprendizados que tive com ele.

Um dos elementos do seu trabalho que achei mais interessante foi o fato de que ele destaca **equipes diferentes dentro de um mesmo projeto para desenvolver métricas diferentes do negócio**. Por exemplo: duas equipes trabalhando paralelamente, uma trabalhando com mecânicas para aquisição, e outra com mecânicas de retenção. Essa é uma questão essencial porque olhando de um ponto de vista estratégico as vezes é difícil priorizar uma métrica de negócio em detrimento de outra, ou métricas em detrimento de funcionalidades de "cosméticas" para usuários. O balanço entre entrega de valor e desenvolvimento de métricas essenciais para o negócio nem sempre é tão claro quanto eu inicialmente pensava. A abordagem que ele me apresentou foi de times menores e mais focados, trabalhando com melhorias incrementais em pontos chave do produto.

Outro aprendizado foi referente à gestão do *roadmap* do produto. O *roadmap* é um planejamento em ordem cronológica das próximas funcionalidades e alterações que serão realizadas no produto em um futuro próximo. Normalmente o *roadmap* é estruturado em forma de calendário ou *gantt*, por conta do foco no aspecto cronológico do projeto.

Esse tópico é delicado porque é necessário buscar um equilíbrio entre um planejamento robusto e uma flexibilidade que permita mudanças de estratégia rápidas na gestão do produto, o que normalmente acontece por meio da repriorização de entregas planejadas no *roadmap*. Segundo Raphael, especialmente se tratando de startups, fazer um *roadmap* com um planejamento de mais de 3 meses é "puro chute", isso porque a estratégia dentro desse período de tempo dificilmente vai permanecer a mesma, sem mudança alguma. Isso significa que não vale a pena fazer um planejamento de *roadmap* que se estende muito para o futuro, mas sim estar sempre re-planejando e re-priorizando as entregas dos próximos 3 meses de forma contínua. Isso não significa que não existe um planejamento à longo prazo, mas sim que o desdobramento da estratégia da empresa em características do produto é mais realista quando planejada sob o olhar de um futuro próximo. Com um *roadmap* de 3 meses é mais fácil prever coisas como tempo de entrega, membros disponíveis para determinado tipo de projeto, e recursos viáveis para a entrega do projeto.

3.2. Imersão

Foi realizada uma imersão de 6 horas em uma startup paulista chamada eduK. A eduK (disponível em: <https://www.eduk.com.br/>) é uma startup do mercado de comércio e educação à distância. Sua atuação consiste no desenvolvimento de cursos técnicos de capacitação em diversas áreas, com o intuito de desenvolver pessoas e prepará-las para o mercado de trabalho, frequentemente com foco em pequenos empreendedores e profissionais autônomos. A visão da eduK é de empoderar pessoas para que elas possam viver trabalhando com suas áreas de interesse e tirar disso o seu sustento. Para isso a startup disponibiliza um portfólio amplo com cursos em formato de vídeo, e conta com uma estrutura robusta de estúdios e equipes de gravação para produção de conteúdos originais. Ela também tem uma grande estrutura de tecnologia para fornecer esses cursos por meio de um modelo de assinatura pela internet. A equipe total da eduK é formada por cerca de 200 funcionários.

A imersão foi realizada por mim em conjunto com dois outros membros da equipe do Ribon: um desenvolvedor e uma pessoa de negócios e relações institucionais.

A imersão começou com uma conversa com Marco Barón, chefe de cultura da eduK. Nessa conversa fui introduzido à cultura organizacional da startup, que trabalha com um modelo de holocracia. A holocracia é um sistema de autoridade distribuída, ou seja, mais horizontal. Esse modelo propõe regras e ferramentas para que a empresa se organize em ciclos autônomos que têm liberdade tática para atuar seguindo a estratégia da empresa como um todo.

Por conta da holocracia, a eduK possui um foco intenso em autogestão, e isso ficou evidente na etapa seguinte da imersão, em que assisti uma reunião do ciclo de cultura da startup. A própria estrutura da reunião ressalta a ideia de autoridade distribuída. Não existia uma figura centralizadora que mediava a reunião, mas sim 4 membros que tinham papéis distintos que normalmente estariam concentrados no mediador da reunião. Esses papéis eram rotativos, e cada ciclo tinha votações de com regularidades variadas para eleger novas pessoas para realizar essas funções. A estrutura da reunião começava com repasses básicos e análises de dados, e gradualmente se encaminha para discussões de projetos que estavam sendo realizados pelos ciclos. Todos os materiais, documentos e registros de reuniões da eduK ficavam concentrados em uma plataforma chamada Glassfrog, que centralizava todas as informações da empresa.

Foi muito interessante perceber as implicações desse modelo de gestão durante a reunião. Um dos membros presentes era cofundador e CTO (chefe de tecnologia) da empresa, mas membros mais novos da equipe se sentiam confortáveis em criticar suas ideias, e a discussão era construtiva com muito diálogo entre os presentes.

A última etapa da imersão foi uma conversa com 3 membros da equipe de produto da eduK, onde pude perceber melhor como o foco em autogestão impactava o projeto de um produto digital. Acho que um dos fatores mais interessantes que pude perceber era como os designers e desenvolvedores da equipe se organizavam em grupos de forma orgânica de acordo com os projetos, diminuindo muito a barreira que existe entre as duas áreas. Essa é uma visão muito importante para o processo de projeto. Membros da equipe de produto de diferentes competências não devem se entender como parte da equipe de design, ou equipe de desenvolvimento, mas sim como membros da equipe de produto sob um único processo unificado.

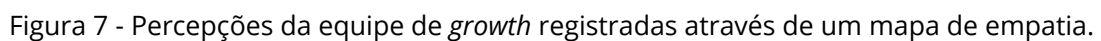
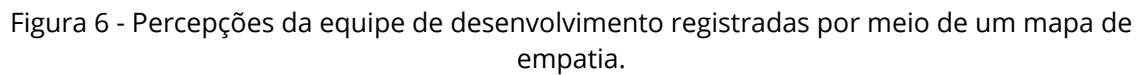
O foco em autogestão que a eduK também apresenta desafios e é um processo em construção. Foi mencionado, por exemplo, um caso em que dois ciclos de produto diferentes estavam trabalhando na mesma solução ao mesmo tempo, sem saber que outra parte da equipe já estava desenvolvendo aquele projeto. O problema foi solucionado com agilidade, cerca de 2 dias. Mesmo assim, é interessante perceber como um foco menor em uma figura que centraliza a liderança apresenta desafios naturais de comunicação entre times diferentes dentro da equipe.

De forma geral, a experiência foi essencial para entender as implicações da autogestão no trabalho de projeto de produto e na coordenação de membros diferentes dentro de uma mesma equipe.

3.3. Entrevistas com a Equipe do Ribon

A participação da equipe do Ribon foi essencial para produzir um modelo de gerenciamento de produto coerente, afinal cada membro da equipe tem competências e particularidades que precisam estar refletidas no processo.

Durante o dia a dia de trabalho na startup, muitas dessas questões são levantadas organicamente durante conversas entre a equipe. Para formalizar esse insights e apresentá-los de forma coerente nesse relatório, foram realizadas 5 entrevistas semi-estruturadas de acordo com o método proposto pelo Grupo Norman Nielsen (2018). Os resultados dessas entrevistas foram formalizados em 2 mapas de empatia (figuras 6 e 7) de acordo com o método proposto pela mesma fonte. Esses mapas representam as percepções de membros do time de desenvolvimento e growth do Ribon:



serviço. Por isso, é importante que os designers realizem entregas completas para os desenvolvedores.

Depois que o processo de *hand-off* acontece, qualquer alteração no modelo de uma funcionalidade tem um custo de tempo muito mais alto para ser realizada. Os desenvolvedores expressaram essa necessidade citando casos em que um leiaute incompleto foi entregue, sendo necessário que eles interrompessem a implementação até ter o leiaute completo em mãos. Ficou clara a necessidade de ter um método mais formal e bem estruturado para *hand-off*, que ajude os desenvolvedores a entenderem uma funcionalidade de forma mais intuitiva e que leve em conta "fluxos alternativos" de usabilidade. Isso significa considerar durante o projeto de uma funcionalidade todos os possíveis fluxos de usabilidade dependendo de condições fora do controle da equipe. Por exemplo: o que acontece se um usuário tentar realizar uma ação dentro da plataforma, mas não tiver acesso à internet no momento? Como deve ser a mensagem de erro mostrada para o usuário? Isso é um exemplo de fluxo que só aparece por conta de um problema de tecnologia (falta de acesso à internet), mas que precisa ser solucionado sob a ótica do design, que vai informar qual é a melhor forma de tratar essa situação levando em conta o ponto de vista do usuário.

Outro problema abordado pelos desenvolvedores foi a falta de pontos de contato mais frequentes com designers e profissionais de *growth*. Equipes de desenvolvimento naturalmente se afastam de outras áreas da empresa por conta da natureza específica do seu trabalho. Existe uma barreira técnica e de linguagem para realizar esse tipo de trabalho, e por isso os desenvolvedores do Ribon acabaram se consolidando em um grupo um pouco mais fechado, que interage menos com outras áreas durante o dia a dia no trabalho.

Ao mesmo tempo, eles tem vontade de estar mais incluídos no processo de projeto em geral. Eles gostariam que a formação de equipes para realizar tarefas específicas acontecesse de forma mais orgânica e interdisciplinar, sem muitos atritos e dificuldades de comunicação mesmo com mais pessoas na empresa.

De forma geral, é possível perceber uma necessidade grande de incluir desenvolvedores de uma maneira mais profunda no processo de projeto de funcionalidades, de forma que eles entendam com mais clareza o propósito por trás de cada projeto, e tenham menos dúvidas sobre fluxos de usabilidade durante a implementação.

A área de *growth* tem algumas propostas de melhoria parecidas: entender os fluxos de usabilidade de cada funcionalidade pela ótica do

usuário também seria muito útil para mapear eventos com mais clareza e identificar oportunidades de incluir mecânicas de aquisição e retenção. Embora considerem que tenham um canal de comunicação aberto com a equipe de design, ainda sentem falta de serem mais demandados pelos designers principalmente no que diz respeito à análise de dados. Esse é um processo essencial para projetar de forma mais coerente, e *growth* tem uma capacidade alta para fornecer *insights* com base em dados que nem sempre é aproveitada pelo restante da equipe.

Outra preocupação levantada foi a agilidade da equipe de produto. Para realizar experimentos de *growth* é importante que a equipe produza testes simples de forma rápida. Assim é possível realizar mais ciclos de testes em um período mais curto de tempo, alcançando uma maturidade maior em uma alteração proposta. Na atuação do Ribon como ela acontecia no momento, a equipe não conseguia realizar esse tipo de entrega simples e rápida, frequentemente estimando errado o tempo de execução necessário para tarefas e demorando mais do que o planejado.

Por fim, a área de *growth* sente necessidade de entender com mais clareza o propósito ou as hipóteses iniciais por parte da equipe de design. Um entendimento melhor desses elementos facilitaria um acompanhamento melhor dos dados após uma implementação de determinada funcionalidade. Seria mais fácil por exemplo determinar se uma funcionalidade foi bem sucedida ou não de acordo com uma métrica de sucesso guia para aquele projeto específico.

Os aprendizados obtidos durante as entrevistas e a síntese em forma de mapas de empatia também foram organizados em temas como demonstrado na figura 8:



Figura 8 - Agrupamento das ideias e insights obtidos por meio dos mapas de empatia.

Insights de naturezas parecidas foram agrupados de acordo com 7 temas, que indicam possíveis pontos de melhora no modelo de gerenciamento de produto. De forma resumida, esses pontos são:

Handoff:

Foi identificada a necessidade de uma entrega mais clara por parte dos designers no *hand-off*. Essa entrega deve mais completa e abranger fluxos de usabilidade alternativos.

Registo e documentação:

O Ribon teve um aumento recente no número de membros da equipe, mas o processo de documentação não acompanhou esse ritmo de crescimento. É necessário centralizar melhor informações sobre o produto para que não haja nenhuma perda e para que elas fiquem acessíveis para toda a equipe.

Agilidade em testes:

A equipe do produto do Ribon tem dificuldade de estimar com precisão o tempo necessário para execução de tarefas. Além disso, a equipe não consegue executar testes rápidos no produto.

Acompanhamento de resultados:

A equipe de produto pode acompanhar resultados de uso de funcionalidades com uma qualidade e periodicidade maior. A expectativa é que isso aumente a qualidade das entregas da equipe, fazendo com que ela trabalhe com um foco em revisões incrementais de funcionalidades já implementadas.

Alinhamento:

Existe um claro desalinhamento entre áreas diferentes dentro da equipe de produto, e foi levantado que funcionalidades em que essas diferentes áreas interagiram com mais frequência e profundidade geraram entregas melhores.

Visibilidade e Transparência:

O processo de projeto de uma funcionalidade não é claro para todos na equipe. É importante esse processo seja visível e acessível para todos para que mais pessoas da equipe possam contribuir para o processo.

Previsibilidade e Causalidade:

Nem sempre é claro como uma funcionalidade impacta a experiência dos usuários com o produto. Também não são claras quais são os propósitos e hipóteses por trás de cada funcionalidade e alteração implementadas no produto.

Essas foram as principais informações levantadas pela equipe de produto do Ribon que serviram para uma estruturação de um processo de gerenciamento de produto mais maduro.

4. PROPOSTA DE SOLUÇÃO

Todas as informações levantadas durante a etapa de pesquisa foram importantes para proposta de modelo de gerenciamento de produto que está sendo desenvolvida no Ribon atualmente. A implementação de um novo modelo não aconteceu em sua totalidade de uma vez só, mas sim de forma iterativa. Diferentes partes e processos do gerenciamento de produto foram melhorados de forma separada de acordo com os aprendizados, algumas dessas partes passaram por mais de uma versão até chegar no modelo atual. O conjunto dessas partes representa o modelo de gerenciamento de produto utilizado no Ribon, que também poderia ser aplicado em qualquer outra startup em um estágio semelhante.

O processo desenvolvido contém quatro grandes etapas: **A - Pré-Projeto, B - Design, C - Desenvolvimento e D - Avaliação**. Cada uma dessas etapas tem suas particularidades e uma importância para o processo como um todo. Elas também são compostas por práticas e ferramentas diferentes. Entre estas, escolhi quatro artefatos para destacar e aprofundar neste projeto, já que elas representam um produto importante dos aprendizados da pesquisa realizada nesse projeto. Esses artefatos são: Roadmap, Documento de *Hand-off*, Pipe de Produto, e Documento Central de Produto.

4.1. Panorama Geral do Modelo de Gerenciamento de Produto do Ribon

Como mencionado, o Modelo de Gerenciamento de Produto do Ribon é dividido em 4 etapas: Pré-Projeto, Design, Desenvolvimento e Avaliação. Ao final de cada etapa acontece uma reunião de alinhamento entre membros da equipe para certificar que o projeto cumpriu todos os requisitos da etapa anterior e pode progredir no processo definido. Apresento um detalhamento maior dessas quatro etapas, na ordem cronológica em que ocorrem durante o processo de projeto de uma funcionalidade.

A - Pré-Projeto

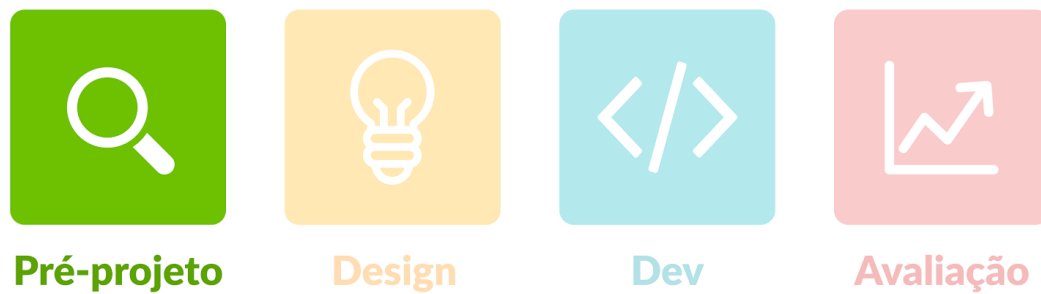


Figura 9 - Etapa de Pré-Projeto

A etapa de pré projeto (figura 9) diz respeito à parte do processo que acontece antes da equipe realmente se comprometer com um projeto específico e começar a desenvolvê-lo. Na prática, isso se traduz em gerenciar um banco de ideias e hipóteses para funcionalidades ou alterações que podem fazer parte do produto no futuro. Esse tipo de ideia pode vir por meio de três canais: pesquisas de design, dados de uso do produto, ou feedback de usuários.

Pesquisas de design acontecem de forma contínua no dia a dia do Ribon para fins diferentes. Elas podem ser realizadas para se projetar uma funcionalidade, para avaliar a satisfação dos usuários com um determinado aspecto do produto, ou até para entender se os anunciantes que patrocinam a plataforma estão satisfeitos com o serviço oferecido, por exemplo. Esse tipo de pesquisa normalmente proporciona um amplo leque de aprendizados, que nem sempre estão restritos ao seu objetivo inicial. Isso acaba gerando um banco de idéias que são interessantes, mas não são o foco atual da atuação da equipe.

Outro canal são dados de uso do produto. Por meio da análise de dados de eventos cadastrados no produto, é possível identificar tendências, oportunidades ou desafios relacionados à experiência dos usuários com a plataforma. Esse processo também gera ideias e hipóteses de melhorias para o produto.

Por fim, um canal que é muito importante e também é um meio para receber novos insights é o feedback de usuários. Esse feedback pode vir pessoalmente à um dos membros da equipe, ou de forma digital por meio de mídias sociais da empresa ou pelo chat disponível no aplicativo, em que

usuários tiram dúvidas e fazem sugestões para a equipe do Ribon. Desde o início do projeto, a empresa sempre buscou manter esse canal de comunicação direto com usuários, e por meio dele muitas propostas de melhoria são apresentadas.

Entre informações recebidas por esses 3 canais, são geradas inúmeras hipóteses de melhorias para o aplicativo. O desafio da etapa de Pré-projeto é justamente gerenciar, aperfeiçoar e priorizar essas hipóteses para que elas se tornem futuras características do serviço do Ribon e entreguem valor para a base de usuários.

Um banco de hipóteses é mantido em um formato simples e menos trabalhado por meio de uma lista em um documento arquivado online. Essas ideias realmente passam por um processamento mais rigoroso quando são alocadas no *Roadmap* do Ribon, indicando a intenção de que elas sejam implementadas em um futuro próximo no produto.

Quando uma dessas ideias vai começar a ser executada pela equipe, acontece o primeiro alinhamento do processo: a reunião de kick-off. O kick-off serve para que toda a equipe de produto se alinhe com relação ao objetivo e às expectativas para aquela funcionalidade proposta, e é o pontapé inicial do projeto. Nessa reunião também é feito um levantamento prévio do que a funcionalidade vai exigir em termos de tecnologia, e de como a equipe pode acompanhar os seus dados e entender se ela foi bem sucedida ou não. Durante o kick-off é preenchido em conjunto um formulário inicial no Pipefy (figura 10), a ferramenta utilizada para acompanhar todas as funcionalidades que estão em desenvolvimento simultaneamente pela equipe. Nesse formulário ficam registrados o nome do projeto, a hipótese que ele pretende testar e quais métricas devem ser acompanhadas para avaliar sua efetividade.

Pipe de Produto (v2)
Form Share

Due date Assignees Attachments +

*Nome do projeto (feature ou ajuste)
Comentários nas Histórias

Hipótese ⓘ
Quando eu estou lendo histórias do Ribon, eu quero interagir e dar minha opinião sobre esse assunto para que eu possa me conectar e conversar com outras pessoas à respeito.

*Essa feature impacta em:
☒ Aquisição
☒ Retenção
[Add new label](#)

[Click here to customize this form](#)

Figura 10 - Formulário de inicial para cadastro de uma funcionalidade nova no Pipefy

B - Design

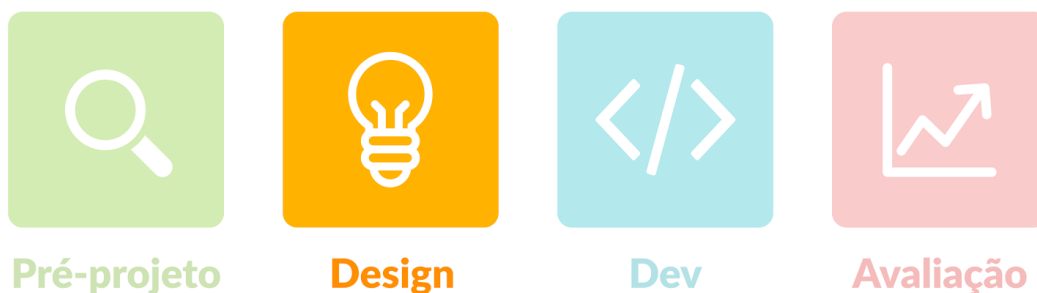


Figura 11 - Etapa de Design

Com o kick-off realizado, a equipe de design está livre para começar o desenvolvimento da funcionalidade (figura 11). O processo de projeto normalmente segue a lógica do Double Diamond como descrito anteriormente. Nesse momento os designers realizam pesquisas de busca de referências e similares, fazem entrevistas com usuários, e começam o desenvolvimento de protótipos para a funcionalidade. No início os protótipos são desenhos simples feitos em papel (figura 12), mas que já podem ser utilizados para confeccionar um protótipo. Os protótipos vão gradualmente avançando em nível de fidelidade até chegar à um leiaute final. A cada nova

etapa o protótipo é apresentado informalmente para membros da equipe do Ribon a fim de coletar percepções sobre o que foi desenvolvido. Os protótipos também são utilizados para realizar testes de usabilidade com usuários do aplicativo. Os aprendizados de cada teste são utilizados para direcionar uma próxima versão, até que o projeto alcance um nível de maturidade razoável. Em geral, a equipe de design consegue realizar uma média duas rodadas de teste de usabilidade por semana, dependendo do nível de demandas. Cada rodada de teste de usabilidade conta com com um mínimo de cinco testes com pessoas diferentes.

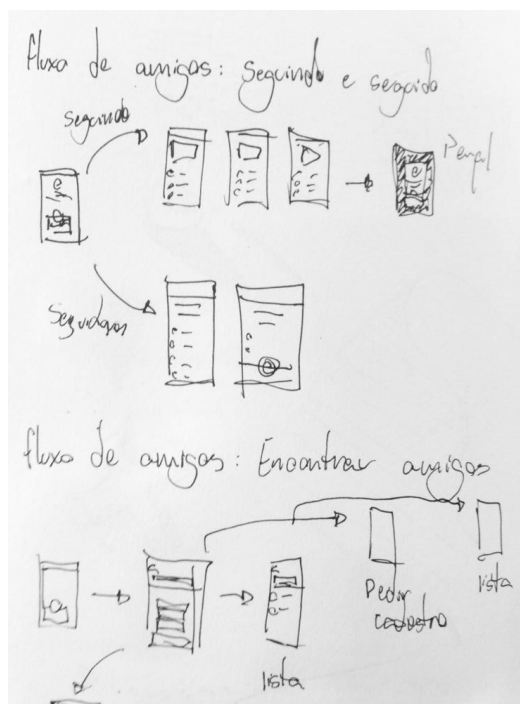


Figura 12 - O projeto de novas funcionalidades normalmente começa com desenhos de baixa fidelidade feitos em papel.

Quando os designers consideram que o projeto atende às necessidades inicialmente colocadas pela equipe de produto e pelos usuários, ela é desenvolvida em formato de leiaute final na ferramenta de design de interfaces Figma (figura 13). O leiaute é composto por um conjunto de telas que demonstram graficamente determinada funcionalidade. Nesse momento, essas telas já estão compostas com os componentes da interface em suas dimensões e características reais, da forma com que elas devem ser renderizadas no aplicativo depois de implementadas.

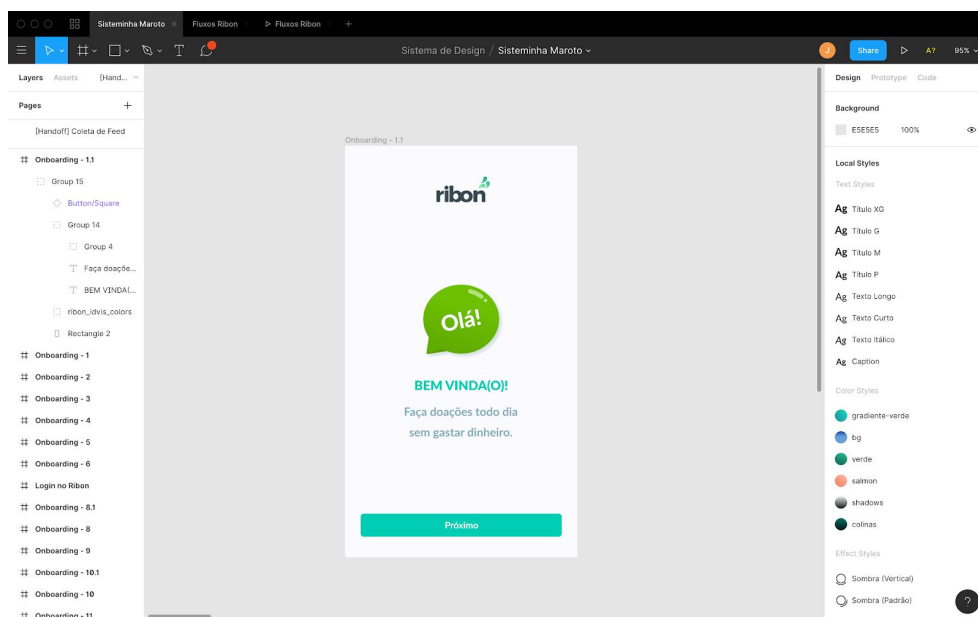


Figura 13 - Figma, software utilizado para confecção dos leiautes.

Com o leiaute pronto, também é uma possibilidade que um protótipo definitivo seja desenvolvido com a finalidade de servir como instrumento para comunicar com mais facilidade o projeto da funcionalidade para a equipe. Isso normalmente acontece quando uma funcionalidade possui um nível de complexidade mais alto, com muitos fluxos de usabilidade.

Quando o trabalho dos designers termina, são realizadas duas etapas de alinhamento: a Aprovação da Funcionalidade e o *Hand-off*.

A aprovação de Funcionalidade é um alinhamento com a equipe completa do Ribon em que a funcionalidade é apresentada e considerações finais são colocadas por todas as áreas da equipe. É nesse momento que os desenvolvedores podem apontar um possível fluxo de usabilidade alternativo que não foi mapeado pelos designers, por exemplo. Quando todas as considerações são colocadas, a equipe de design pode realizar pequenas revisões para afinar detalhes antes da funcionalidade seguir para implementação.

Com esse alinhamento concluído, a equipe de design se reúne com os desenvolvedores que vão trabalhar na implementação da funcionalidade em uma reunião de *Hand-off*. Esse alinhamento tem um caráter mais técnico, onde a funcionalidade é revisada com mais detalhamento, focando menos em proposta de valor, e mais em detalhes de desenvolvimento como transições entre telas, possíveis animações, etc. Fundamentalmente, o *Hand-off* serve como uma entrega da equipe de design para a equipe de desenvolvimento,

passando a responsabilidade de uma funcionalidade específica de uma equipe para a outra.

C - Desenvolvimento

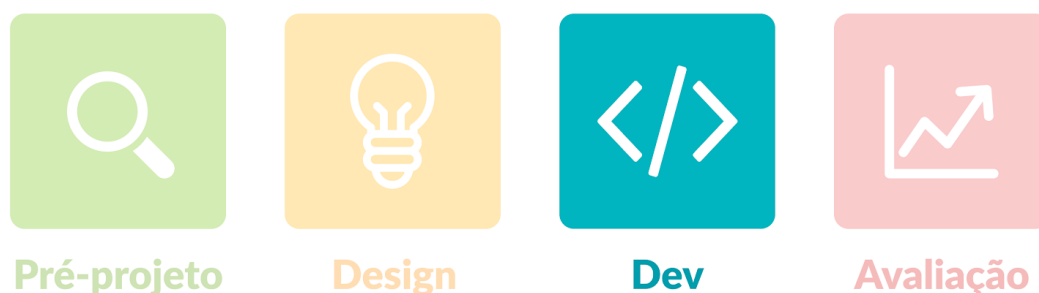


Figura 14 - Etapa de Desenvolvimento

A partir do leiaute formalizado em documento de Hand-off, os desenvolvedores começam o trabalho de implementação da funcionalidade (figura 14). O número de desenvolvedores alocado para realizar a implementação depende do nível de complexidade da funcionalidade, mas em geral é preferível que eles trabalhem no mínimo em duplas. Nesse modelo, um dos desenvolvedores trabalha no *back-end*, ou seja, faz a estruturação da funcionalidade no sistema geral do Ribon. Isso inclui estruturação da API do Ribon para receber a nova funcionalidade, ou integrações com outras APIs, como a do Facebook, por exemplo. A outra parte da dupla trabalha no *front-end*, o que significa que ele é responsável por implementar a interface gráfica que vai ser utilizada por usuários para interagir com aquela funcionalidade.

Independente do número de desenvolvedores alocados e da forma de divisão de tarefas escolhida, os desenvolvedores fazem alinhamentos informais diários com os designers a fim de certificar que a implementação está seguindo o projeto original da funcionalidade.

Quando o trabalho de implementação está completo, é realizado o alinhamento dessa etapa. Esse alinhamento é chamado de Pré-Lançamento, e é realizado por toda a equipe de produto com o objetivo de certificar que a funcionalidade está pronta para ser lançada para a base de usuários. Nesse momento, os designers fazem as pontuações finais sobre a implementação e qualquer detalhe gráfico que precise ser resolvido, e os profissionais de

growth se certificam de que a funcionalidade está com os eventos corretos cadastrados para acompanhamento dos dados à longo prazo.

Depois desse alinhamento final, os desenvolvedores fazem uma nova versão do produto que inclui essa melhoria mais recente, e disponibilizam essa atuação para a base de usuários. Dessa forma, é realizado o processo de *release*, e os desenvolvedores consideram seu trabalho entregue.

D - Avaliação

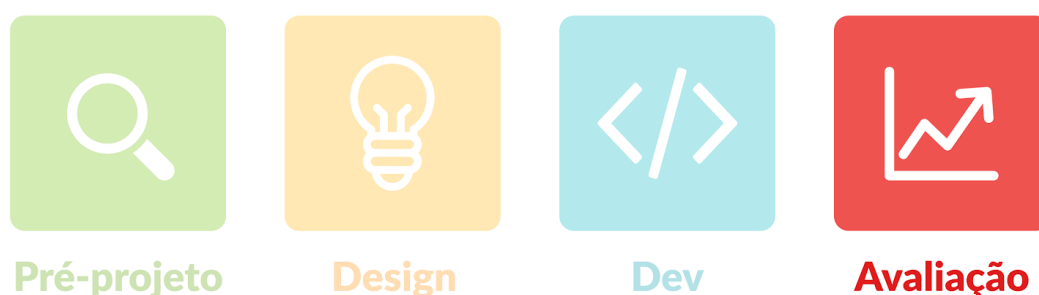


Figura 15 - Etapa de Avaliação

A última etapa do processo de gerenciamento de produto é a avaliação (figura 15), em que funcionalidades já implementadas são acompanhadas semanalmente para que a equipe tenha um entendimento claro da sua performance. A maior parte de trabalho dessa etapa é responsabilidade da área de *growth*, que analisa os dados de uso do aplicativo e reporta para a equipe regularmente.

É nessa etapa que a equipe vai entender se uma funcionalidade foi bem sucedida ou não, baseando-se no objetivo inicial desenhado para o projeto e nas métricas que foram definidas para serem acompanhadas.

Nesse momento a equipe pode considerar que a funcionalidade está atendendo ao seu propósito de forma satisfatória, ou que ela ainda precisa de ajustes para ser realmente efetiva. Caso ela necessite de uma revisão, ela retorna para a etapa de Design no modelo de gerenciamento de produto, onde ela pode sofrer desde uma leve revisão até uma reformulação completa. A necessidade desse tipo de revisão não necessariamente indica um erro de projeto. Pelo contrário, os ajustes incrementais no produto são esperados e necessários para atingir uma maturidade na oferta de valor para os usuários,

e isso está em conformidade com a natureza iterativa da prática projetual em startups.

4.2. Detalhamento de Artefatos no Modelo de Gerenciamento de Produto

Como mencionado, decidi detalhar quatro artefatos específicos que são utilizados no modelo de gerenciamento de produto desenvolvido. Esses artefatos específicos foram escolhidos para detalhamento porque passaram por um processo de desenvolvimento mais robusto durante a realização do projeto, com mais iterações e mais aprendizados internalizados. Esses quatro artefatos serão detalhados seguindo a ordem cronológica em que foram apresentados no tópico anterior, e são: *Roadmap*, Documento de *Hand-off*, Pipe de Produto, e Documento Central de Produto.

Roadmap

O *Roadmap* (figura 16) serve para planejar as melhorias que serão implementadas em um determinado produto em um futuro próximo. Normalmente ele é estruturado em um formato cronológico claro como em um calendário, por exemplo.

No início do desenvolvimento do projeto, o Ribon possuía um *Roadmap* que era estruturado e priorizado pela equipe de design, e mantido no Google Agenda.

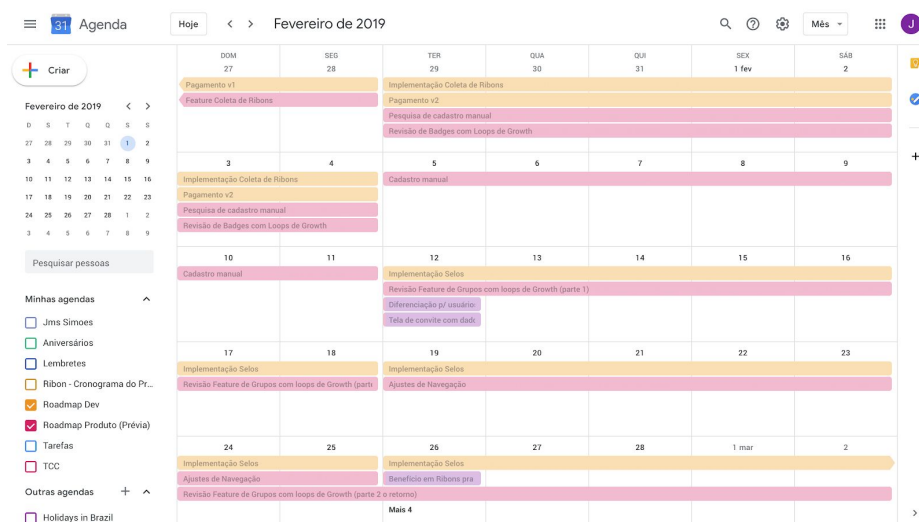


Figura 16 - Primeira versão do *Roadmap*

Essa versão do *Roadmap* representava um avanço considerável em relação à forma com que o planejamento era feito anteriormente, onde o processo de tomada de decisão era mais informal e o planejamento não era documentado. Isso permitia que a equipe de design conseguisse operar com mais previsibilidade com relação ao futuro do produto, entendendo com mais clareza quando estariam disponíveis para começar o projeto de novas funcionalidades. Mesmo assim, a equipe de produto ainda não mantinha um ritmo regular de *releases*, e funcionalidades projetadas ainda demoravam mais do que o planejado para chegar às mãos dos usuários.

Por meio da pesquisa realizada durante o projeto, principalmente a conversa com o Raphael Farinazzo, um erro de execução desse planejamento ficou claro. O *roadmap* foi desenvolvido pela equipe de design de uma forma isolada, e descrevia os tempos das entregas da equipe de design, quando na verdade, ele deveria ser realizado sob a visão dos usuários, ou seja, através de *releases*. As *releases* definem o momento de entrega de valor para usuários, sendo a única data realmente relevante para ser acompanhada pela equipe de produto como um todo. Fazer um novo *roadmap* com base no momento da entrega de *releases*, e não no momento de entregas de áreas específicas, exigiu um novo nível de alinhamento da equipe de produto. Um novo modelo de *roadmap* foi desenvolvido. Embora a área de design ainda fosse a responsável por esse planejamento, agora ele levaria em conta particularidades do processo de outras áreas como desenvolvimento e growth.

No modelo atual (figura 17), incluir uma nova funcionalidade no *roadmap* exige um nível de alinhamento alto principalmente com os desenvolvedores, que tentam prever o tempo necessário para implementar cada funcionalidade e o tempo para que ela esteja disponível em *release*. Dessa forma o *roadmap* leva em conta o tempo necessário para o desenvolvimento de cada funcionalidade como um todo, começando no Pré-Projeto e terminando apenas no momento de *release*. As datas de *release* passaram a ser pontuadas no *roadmap* com destaque como forma de denotar prazos finais para ter determinada funcionalidade completamente implementada.

O *roadmap* também passou de um formato de calendário para a ferramenta Team *Gantt*. Essa ferramenta ajuda a equipe de produto a priorizar o planejamento com mais agilidade, e a prever como o atraso em uma *release* afeta as demais no futuro, em formato de cascata.

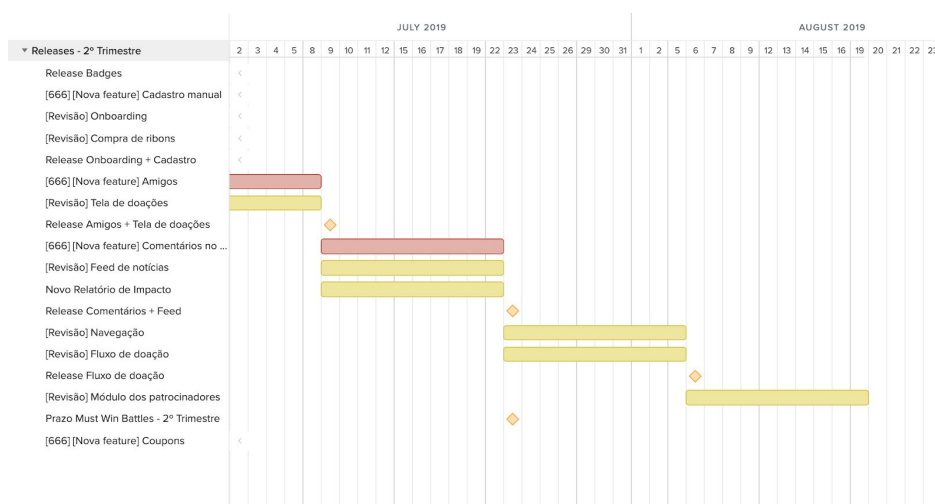


Figura 17 - Segunda versão do *Roadmap*.

O atual modelo de *Roadmap* se preocupa com o resultado final das funcionalidades e quando elas estão disponíveis para usuários. Isso auxiliou a equipe a ter uma visão mais realista a respeito do planejamento futuro do produto.

Documento de *Hand-off*

Antes do desenvolvimento do projeto o *Hand-off* dos leiautes feitos pelos designers para os desenvolvedores era realizado de forma não estruturada (figura 18). O arquivo onde o leiaute havia sido confeccionado era compartilhado com os desenvolvedores, que podiam visualizar as telas desenvolvidas. Os designers então explicavam como a funcionalidade deveria funcionar, e os desenvolvedores podiam começar a implementação.

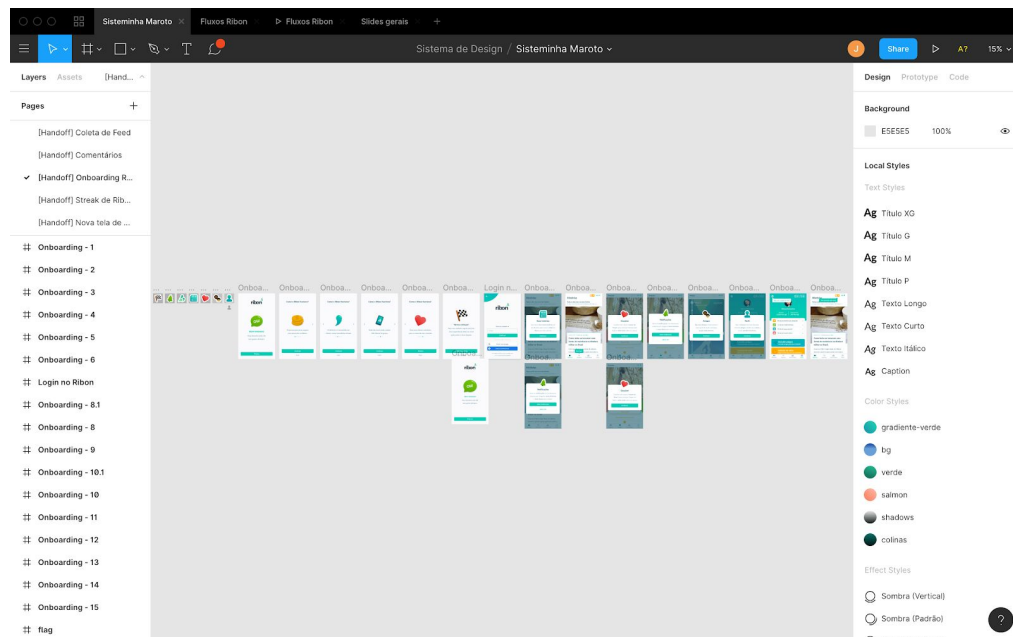


Figura 18 - Antes do projeto o *Handoff* era feito de forma desestruturada, em que as telas não eram organizadas formalmente, e não havia nenhuma descrição contextual.

Esse modelo é utilizado por muitas empresas, mas tem alguns problemas fundamentais. Os desenvolvedores não participam ativamente da pesquisa e projeto de design da funcionalidade, e portanto, não tem um entendimento profundo de como ela deve funcionar. Além disso, os designers têm mais dificuldade em prever fluxos alternativos de usabilidade de uma funcionalidade enquanto estão projetando porque não possuem um embasamento técnico de desenvolvimento. Isso faz com que eles não consigam prever com clareza momentos de falha de tecnologia que precisam ser comunicados para o usuário. Frequentemente novas telas precisavam ser projetadas para levar em consideração esses momentos, e executar esse trabalho de design durante a implementação em código de uma funcionalidade adicionava atrito e desafios de comunicação ao projeto.

Durante a pesquisa, ficou claro que realizar entregas de *Hand-off* de forma mais estruturada era necessário, e organizar as telas em forma de fluxo de usabilidade seria essencial para que uma funcionalidade fosse mais facilmente entendida por toda a equipe mesmo antes da implementação.

Foi elaborado então um formato padrão de documento de *Hand-off* (figura 19), que não entrega as telas separadas de forma desestruturada, mas sim descrevendo o caminho que um usuário faria por meio dessas telas. Cada funcionalidade pode ter diversos fluxos possíveis, e todos estão mapeados no seu documento de *Hand-off*.

Fluxo da tela de Amigos

Quando estou usando o Ribon com amigos eu quero:

1. Criar e customizar o meu perfil para que saibam que sou eu;
2. Saber quanto eu e meus amigos doamos, sentir que estamos progredindo;
3. Seguir novas pessoas, saber quem eu sigo, e quem está me seguindo;
4. Encontrar amigos que já usam o Ribon, ou convidar quem não usa;

Para que eu me sinta conectado com meus amigos através do Ribon.

Ultima atualização: 02/06/19

Figura 20 - Job Stories em um documento de Hand-off

Fluxo da tela de Amigos

Quando estou usando o Ribon com amigos eu quero:

1. Criar e customizar o meu perfil para que saibam que sou eu;
2. Saber quanto eu e meus amigos doamos, sentir que estamos progredindo;
3. Seguir novas pessoas, saber quem eu sigo, e quem está me seguindo;
4. Encontrar amigos que já usam o Ribon, ou convidar quem não usa;

Para que eu me sinta conectado com meus amigos através do Ribon.

Última atualização: 02/06/19

- 1: Quero criar e customizar meu perfil para que saibam quem eu sou.

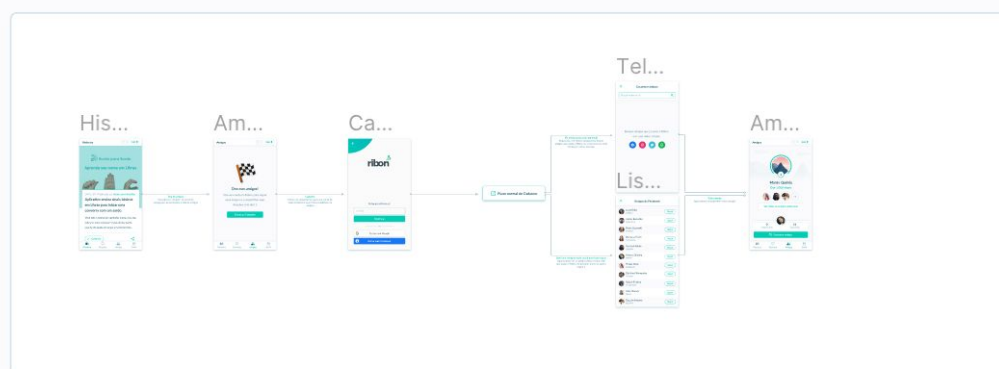


Figura 21 - O documento de *handoff* completo é composto por uma parte inicial com Job Stories que pode informar um ou mais possíveis fluxos de usabilidade, que são mapeados no documento.

O documento de Hand-off foi essencial para garantir que o trabalho desenvolvido pela equipe de design fosse transparente para todos os membros da empresa. De um ponto de vista técnico, ele foi responsável por reduzir consideravelmente as dúvidas dos desenvolvedores durante o processo de implementação, tornando o processo mais ágil e menos incerto.

Kanban de Produto

O pipe de produto (figura 22) veio da necessidade de uma ferramenta central para acompanhar o progresso de todas as funcionalidades desenvolvidas pela equipe de produto. Ele tem um formato simples de *Kanban* que reflete a estrutura do Modelo de Gerenciamento de Produto, onde cada coluna representa uma etapa do processo e seus respectivos momentos de alinhamento.

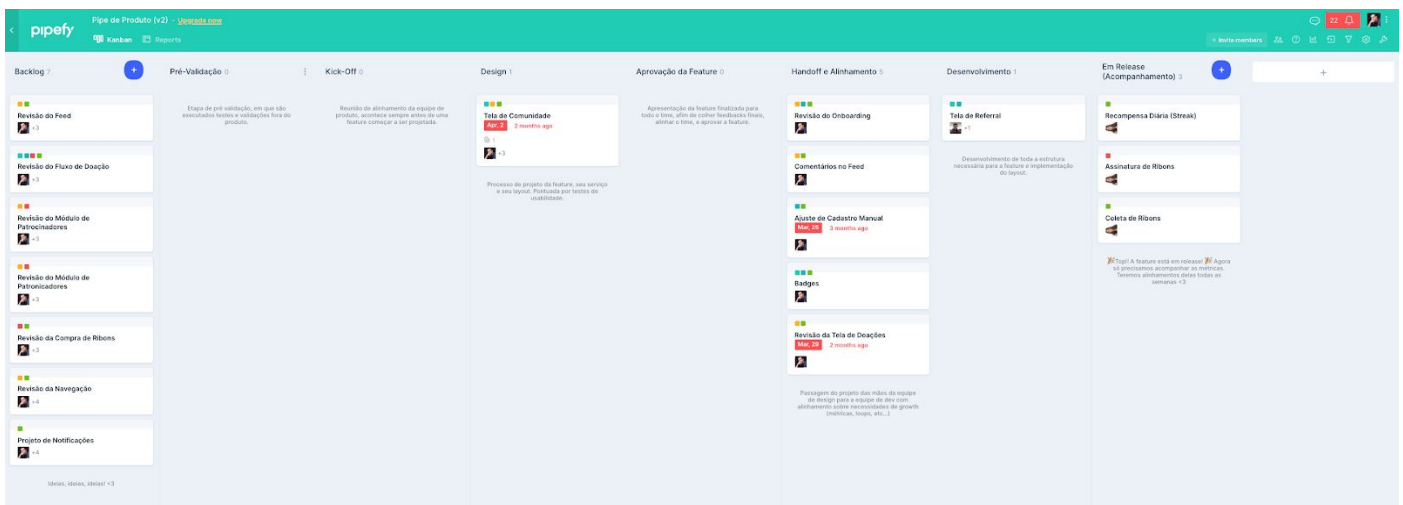


Figura 22 - Kanban de acompanhamento de funcionalidades no Pipefy.

O *Kanban* foi executado na ferramenta Pipefy, que pode ser acessada por qualquer membro da equipe que quiser saber em que estágio do processo uma funcionalidade se encontra. Ele também registra o nome e hipótese de cada funcionalidade, sendo possível entender como ela vai impactar em termos de métricas no produto.

Outro benefício de se visualizar o trabalho da equipe de produto utilizando a ferramenta Pipefy e identificar possíveis gargalos no processo onde funcionalidades costumam demorar mais para serem desenvolvidas, ou quais estão próximas de serem finalizadas e podem ser incluídas na release mais próxima.

Documento Central de Produto

O gerenciamento de produto exige um rigor com registro de dados e documentação do desenvolvimento do produto. Quanto maior a equipe, mais difícil é de transmitir essas informações. Durante o desenvolvimento desse projeto o tamanho da equipe de produto do Ribon mais que dobrou, e perdas de informação e problemas de comunicação começaram a se tornar mais comuns.

Métodos mais clássicos de gerenciamento de produto, como o descrito por Ben Horowitz (Good Product Manager, Bad Product Manager, 1997), possuem uma ênfase grande em documentação, e frequentemente propõe o uso de um modelo chamado Product Requirements Document (PRD). O PRD é um documento centralizador de todas as informações relevantes de um produto, que deve ser atualizado no mínimo semanalmente por um gerente de produto.

O modelo foi adotado pelo Ribon, com algumas adaptações (figura 23). No seu formato atual, o Documento Central de Produto é separado por funcionalidades e contém informações gerais como data de lançamento, dados coletados e hipótese. De acordo com o foco em autogestão proposto pela empresa, ele é atualizado por todos os membros da equipe de acordo com o momento em que a funcionalidade se encontra dentro do Modelo de Gerenciamento de Produto.

Os principais benefícios percebidos pela equipe com a adoção do documento são a centralização de informações sobre o produto, e uma documentação abrangente que possibilita que a equipe entenda com mais clareza o histórico de lançamento de funcionalidades do produto, permitindo uma análise de dados mais profunda.

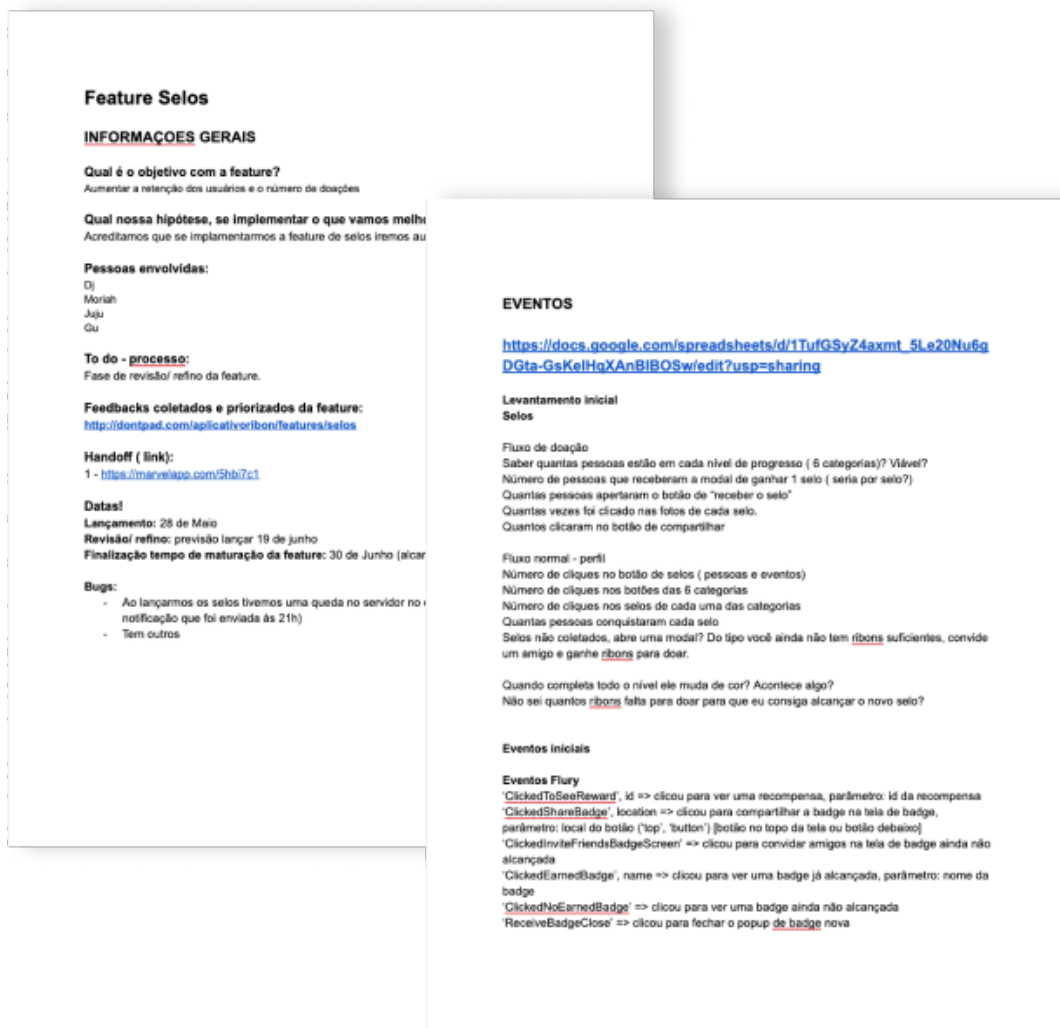


Figura 23 - Páginas do Documento Central de Produto.

5. CONSIDERAÇÕES FINAIS

O projeto desenvolvido permitiu que a equipe do Ribon aumentasse consideravelmente a qualidade nas entregas de produto. Tendo em vista que o formato da empresa exige entregas rápidas e regulares, o Modelo de Gerenciamento de Produto é uma ferramenta importante para manutenção da qualidade do processo de projeto de novas funcionalidades.

Os principais resultados aparentes do projeto foram divididos em 3 áreas:

1- Alinhamento e Comunicação

O modelo proposto teve um impacto positivo significativo na comunicação da equipe de produto, e foi fundamental para que ela se entenda como uma equipe única, apesar das áreas de atuação diferentes de seus membros. Ele foi importante para tornar o trabalho da equipe mais acessível não só para os seus membros como para todos dentro da startup.

2 - Dados e Informações

O modelo proposto tem uma ênfase na documentação e tratamento de informação. Atualmente é possível entender melhor o propósito por trás de cada melhoria do produto, além de trazer mais dados para o processo de projeto em todas as suas etapas, possibilitando tomadas de decisão mais embasadas e mais qualidade nas entregas.

3 - Priorização e Agilidade

O modelo proposto, apesar de formalizar o processo, ainda permite flexibilidade no planejamento além de possibilitar que a equipe priorize suas tarefas com mais segurança. Ele também teve um impacto positivo na agilidade da equipe de produto, que consegue realizar testes e entregas mais rápidas. Poucos dias antes da entrega deste relatório, uma funcionalidade teste foi projetada e implementada em *release* em menos de 5 dias, um bom resultado que foi muito comemorado pela equipe.

Assim como uma startup nunca considera o seu produto oferecido como completamente "pronto", a gestão e a operação da empresa também estão em constante evolução. O Modelo de Gerenciamento desenvolvido durante o projeto apresenta muito espaço para melhora, e o projeto me ajudou a entender não só como estruturar esse tipo de processo em startups, mas também quais são os pontos de atenção e valores norteadores para que ele continue evoluindo.

REFERÊNCIAS BIBLIOGRÁFICAS

Aplicativo **Ribon**. Disponível em ribon.io

CAGAN, Martin, Silicon Valley Product Group. **How to Write a Good PRD**.

Disponível em: <https://svpg.com/assets/Files/goodprd.pdf>

CAO, Jerry. **How to Write a Painless Product Requirements Document**.

Disponível em:

<https://medium.com/@uxpin/how-to-write-a-painless-product-requirements-document-508ff6807b4a>

Design Council. **The Design Process: What is the Double Diamond?** 2006.

Disponível em:

<https://www.designcouncil.org.uk/news-opinion/design-process-what-double-diamond>

eduK. Disponível em: <https://www.eduk.com.br/>

GIBBONS, Sarah, Nielsen Norman Group. Empathy Mapping: **The First Step In Design Thinking**. Disponível em:

<https://www.nngroup.com/articles/empathy-mapping/>

HANDLEY, Alexander. **User flow is the new wireframe**. Disponível em:

<https://uxdesign.cc/when-to-use-user-flows-guide-8b26ca9aa36a>

HOROWITZ, Ben. **Good Product Manager, Bad Product Manager**, 1997.

KLEMENT, Alan. **Designing features using Job Stories**. Disponível em:

<https://www.intercom.com/blog/using-job-stories-design-features-ui-ux>

Manifesto Ágil. 2001. Disponível em: <https://www.manifestoagil.com.br/>

Nielsen Norman Group. **User Interviews: How, When, and Why to Conduct Them**. Disponível em: <https://www.nngroup.com/articles/user-interviews/>

LALAUX, Frédéric. **Reinventing Organizations**, 2014.

OSTERWALDER, A ,PIGNEUR, Y, BERNARDA, G , SMITH, A. **Value Proposition Design**, 2014.

PERNICE, Kara, Nielsen Norman Group. User Interviews: **How, When and Why to Conduct Them**. Disponível em:

<https://www.nngroup.com/articles/user-interviews/>

RIES, Eric. **The Lean Startup**, 2011.

ANEXOS

Nas próximas páginas disponibilizo o texto resumido do Manifesto Ágil, que utilizei como referência:

Manifesto para o desenvolvimento ágil de software

Estamos descobrindo maneiras melhores de desenvolver software, fazendo-o nós mesmos e ajudando outros a fazerem o mesmo. Através deste trabalho, passamos a valorizar:

1. Indivíduos e interações mais que processos e ferramentas
2. Software em funcionamento mais que documentação abrangente
3. Colaboração com o cliente mais que negociação de contratos
4. Responder a mudanças mais que seguir um plano

Ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda. Nós seguimos os seguintes princípios:

- Nossa maior prioridade é satisfazer o cliente, através da entrega adiantada e contínua de software de valor.
- Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas.
- Entregar software funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos.
- Pessoas relacionadas à negócios e desenvolvedores devem trabalhar em conjunto e diariamente, durante todo o curso do projeto.
- Construir projetos ao redor de indivíduos motivados. Dando a eles o ambiente e suporte necessário, e confiar que farão seu trabalho.
- O Método mais eficiente e eficaz de transmitir informações para, e por dentro de um time de desenvolvimento, é através de uma conversa cara a cara.
- Software funcional é a medida primária de progresso.
- Processos ágeis promovem um ambiente sustentável. Os patrocinadores, desenvolvedores e usuários, devem ser capazes de manter indefinidamente, passos constantes.
- Contínua atenção à excelência técnica e bom design, aumenta a agilidade.

- Simplicidade: a arte de maximizar a quantidade de trabalho que não precisou ser feito.
- As melhores arquiteturas, requisitos e designs emergem de times auto-organizáveis.
- Em intervalos regulares, o time reflete em como ficar mais efetivo, então, se ajustam e otimizam seu comportamento de acordo.

Escrito por:

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler
James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick
Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas